High Fidelity Facial Animation Capture and Retargeting With Contours

Kiran S. Bhat* Rony Goldenthal† Yuting Ye‡ Ronald Mallet§ Michael Koperwas¶

Industrial Light & Magic

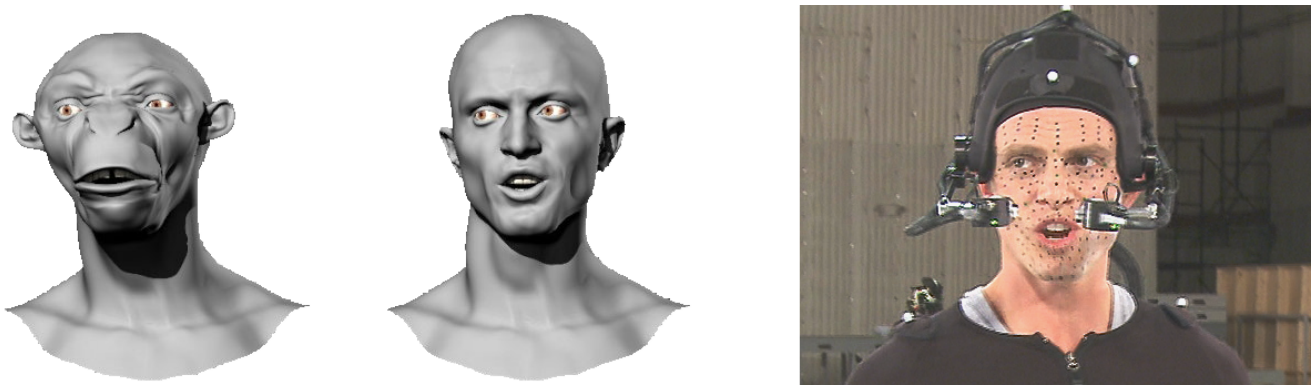


Figure 1: *Our performance-capture approach excels at capturing and retargeting mouth and eyelid motion accurately.*

Abstract

Human beings are naturally sensitive to subtle cues in facial expressions, especially in areas of the eyes and mouth. Current facial motion capture methods fail to accurately reproduce motions in those areas due to multiple limitations. In this paper, we present a new performance capture method that focuses on the perceptually important contour features on the face. Additionally, the output of our two-step optimization scheme is also easily editable by an animator. To illustrate the strength of our system, we present a retargeting application that incorporates primary contour lines to map a performance with lip-sync from an actor to a creature.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.3.3 [Computer Graphics]: Animation—Motion capture

Keywords: facial animation, performance capture, facial retargeting, contours, curves, blendshape, correctives

1 Introduction

Digital characters, both humans and creatures, are becoming prevalent in feature films and video games, oftentimes performing along side human actors on screen. With realistically rendered images, it is a grand challenge to deliver convincing facial animations that can convey emotions and communicate with the audience. Perception studies [Cunningham et al. 2005; Nusseck et al. 2008] show that humans rely heavily on facial expressions in communications. For example, micro-expressions helps to decipher emotions; and lip reading is vital to understand dialogues. Even the slightest dissonance between these facial cues and viewers' expectations may result in false emotions [Ekman 1993]. To date, the art of animating a realistic face remains in the hands of few expert artists. The quality of an animation depends on modelers' ability to craft a large number of blendshapes that capture the most distinctive expressions of the actor, and the animators' ability to animate a complex set of shapes.

Performance capture techniques provide an efficient alternative to create compelling facial animations. Simultaneous capture of an actor's facial and body motion, where an actor is free to move on-set, is an ideal setting for production use. The most common solution is marker-based video systems because they are flexible and reliable. However, the sparse set of markers cannot provide sufficient details to reconstruct the nuances on a realistic face. Areas around the eyelids and the mouth are especially problematic due to frequent self-occlusions seen from the camera. In practice, artists have to manually clean-up the animation, by both sculpting additional blendshapes and carefully adjusting the animation curves. The results are sometimes inconsistent across artists, and difficult to evaluate objectively. Consequently, while background characters can be animated efficiently using performance capture, high quality hero characters are still labor intensive and expensive to produce.

In this work, we present an artist-friendly system that is capable of producing high fidelity facial animations. An important feature of our system is the incorporation of silhouette contours of the eyelids and the inner mouth to reconstruct accurate animation. Given a set of blendshapes and a set of $2D$ features (markers and contours), our system automatically solves for animation curves on the blendshapes and produces an additional *corrective* at each frame of the sequence. The combination of blendshapes and correctives accurately match the movements of the skin, eyes and the inner mouth. While it is relatively straightforward to match fixed features such as markers, it is a challenge to match the occluding contours because

*email: kbhat@ilm.com
†email: ronygold@gmail.com
‡email: yye@ilm.com
§email: rmallet@ilm.com
¶email: mkoper@ilm.com

SCA 2013, July 19 – 21, 2013, Anaheim, California.

the silhouette is constantly changing on the skin. We therefore employ a simple yet effective heuristic to dynamically associate the tracked curves with contours on the mesh at every frame based on visibility information.

While a linear blendshape system with a fixed set of shapes can provide actor-specific spatial nuances such as wrinkles on the forehead and fine-scale folds on the skin, it typically fails to capture subtle and complex nonlinear interactions between different blendshapes. Moreover, it is difficult to match the inner mouth accurately with a relatively small number of blendshapes. Instead of having artists manually sculpt correctives, we solve for these additional corrective shapes on top of the blendshape animation to better match the features at every frame using a Laplacian deformation method [Botsch and Sorkine 2008]. This two-step approach not only adds the needed subtleties to the motion, but also ensures that artists can work with a familiar blendshape system to edit the animation.

We apply our system to the performance of two professional actors with subtle expressions and dialogues. The resulting animation delivers emotions, accurate eye blinks, and readable lips, all from a sparse set of $2D$ features. The otherwise challenging lip rolling motion is made possible in our system with dynamically selected occluding contours. In addition, we also show that an accurate performance of the actor can directly drive a believable talking creature. With a simple heuristic to warp the actor's lip shape to the talking portion of the creature's mouth, a lip-synced creature is readily achieved. Our work showcases the importance of incorporating occluding contours around the eyes and inner mouth, and presents a production tested system for creating high quality facial animation.

2 Related work

Highly detailed facial expressions and performance can be captured by dedicated hardware in controlled environments. 3D scanning techniques such as multi-view videos [Fyffe et al. 2011; Ghosh et al. 2011; Beeler et al. 2011] and structural lights [Zhang et al. 2004; Zhang and Huang 2004; Weise et al. 2009; Li et al. 2009; Mova ; Dimensional Imaging Ltd] are used to capture geometric details of the skin and photometric information. These techniques can be used to create realistic digital doubles [Alexander et al. 2010; Borshukov et al. 2005]. However, the constrained capturing environment is not suitable for productions that require an on-set capture of facial animation and body motion at the same time. Moreover, scanning techniques cannot acquire accurate information around the inner mouth and eyes, which are important for high fidelity performance.

2D video-based techniques are more suitable due to their flexible capture settings. Marker-less techniques rely on texture information (AAM [Cootes et al. 1998], optical flow [Horn and Schunck 1981]) to automatically track features from video [Chuang and Bregler 2002; Black and Yacoob 1995; Covell and Bregler 1996; Baltrusaitis et al. 2012]. Despite the overall high-quality these methods are sensitive to fast motion and lighting variation, which is common in on-set capture. Marker based techniques are preferred in production environment due to their reliability under different circumstances [Williams 1990; Bickel et al. 2007; Huang et al. 2011]. The tracked data is, however, sparse and lacking fine skin detail. In particular, it's difficult to track around the eyelid contour and inner lip contour due to occlusion.

The facial features around the eyes and mouth are especially important for high quality facial animation. Commercial applications such as Live Driver^{TM}[Image Metrics] and HMC & Grabber^{TM}[dynamixyz] track only these regions for digital puppetry, and they obtain impressive results. Early works such as [Terzopoulos and Waters 1993; Bascle and Blake 1998] used contours to track these areas. In our work, we use contours to track the eyelid and the silhouette contour of the inner lip since it also captures lip motion orthogonal to the camera. We also enforce eyelid-cornea contact to help determine the 3D location of the image-based curve.

Editable facial animation is obtained by fitting a facial rig [Orvalho et al. 2012] to the tracked data. Using a facial rig applies prior knowledge that can be used to fill in missing information not captured from the video. For example, designated blendshape rigs are designed to capture the distinctive expressions of an actor [Pighin et al. 1999; Liu et al. 2008; Li et al. 2010; Huang et al. 2011; Seo et al. 2011]. There has been research to apply anatomy-based muscle models [Sifakis et al. 2005] to simulate facial animation. The results are physically plausible, but difficult to control and animate. Data-driven models can be used to train facial shape priors or motion priors [Li et al. 2010; Weise et al. 2011; Ma et al. 2008]. However, the training sessions are intensive and it is difficult to obtain consistent training data. Although our work also builds on a blendshape rig that artists are familiar with, we allow for out-of-model animation that captures nuances without relying on an elaborate set of shapes. In parallel to our work, Li *et al.*[2013] apply a similar approach to face tracking using per-frame correctives. To obtain realtime performance, however, they use fixed contours around the outer mouth instead of the occluding contours of the inner mouth.

Recently, sketch-based interfaces [Lau et al. 2009; Miranda et al. 2011] were applied to control facial expressions, especially of the eye and mouth region. We are inspired by the ease of use of these methods and similarly utilize the eyes and mouth contours to animate the face.

3 Performance capture pipeline

The input to our system are video streams of an actor's facial performance, a neutral mesh of the actor's face, and a set of blendshapes built for the actor. Our system then automatically reconstructs the facial performance on the given mesh to match features captured from the video.

The input videos are recorded from two synchronized HD cameras attached to a helmet worn by the actor (Figure 1). The actor wears stipple makeup to provide sparse dot patterns on the skin . This setting is common in film production because it works with body mocap, provides maximal coverage of the actor's face, and the cameras are static relative to the head. From the video, we can estimate camera parameters and track the dots using an automatic patten tracking algorithm similar to [Bregler et al. 2009]. In addition to the tracked dots, we also manually trace the outline of the upper and the lower eyelids, as well as the silhouettes of the inner lip (fig 4(a)) that are the primary contour lines on the face. The inner lip curves emanate from the corner of the mouth and extend to the center of the mouth.

For an input frame, we reconstruct a mesh that matches the tracked dots and curves by solving two optimization problems. First, we compute blendshape weights to find the optimal shape within a given blendshape subspace. Next, we minimize the remaining matching error by solving for an additional blendshape specific for the given frame. The final result is a blendshape animation that captures nuance expressions with accurate eyelids and inner mouth shape. Animators have the option to edit the resulting animation curves with standard blendshape editing tools.

4 Animation reconstruction

Given a sparse set of $2D$ features as input, our goal is to reconstruct the actor's performance by solving two optimization problems. Since they utilize the same set of constraints, we will first

describe our constraint formulation, followed by the optimization framework.

4.1 Point constraints

The 2*D* markers are tracked automatically from the dot patterns painted on the actor's face. Since their positions are fixed to the skin, we can compute the corresponding points on the actor's neutral geometry from a chosen frame. A point $\mathbf{x}$ on the mesh surface is represented using barycentric coordinates: $\mathbf{x} = \mathbf{A}\mathbf{V}$, where $\mathbf{A}$ is the barycentric coordinates relative to its triangle in matrix form, and $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ is a vector of all vertices $\mathbf{v}_i$ on the mesh.

2*D* markers For a given 2*D* marker $\mathbf{m} = \{m_x, m_y\}$ and the corresponding point $\mathbf{x}$ represented in homogeneous coordinates as $\mathbf{y}$, we define the fitting error as their distance on the image plane.

$$\mathbf{c}_m(\mathbf{V}) = \begin{bmatrix} m_x\mathbf{Q}_2^\top\mathbf{y} - \mathbf{Q}_0^\top\mathbf{y} \\ m_y\mathbf{Q}_2^\top\mathbf{y} - \mathbf{Q}_1^\top\mathbf{y} \end{bmatrix}, \tag{1}$$

where $\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$, and $\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_0^\top \\ \mathbf{Q}_1^\top \\ \mathbf{Q}_2^\top \end{bmatrix}$ is the 3×4 camera projection matrix obtained using bundle adjustment with a standard camera calibration framework.

3D bundles 2*D* Markers at the center of the face can be seen from both cameras. We can therefore estimate their 3*D* positions using the bundle adjustment method. The following 3*D* point constraint is used to fit a bundle $\mathbf{p}$.

$$\mathbf{c}_p(\mathbf{V}) = \mathbf{x} - \mathbf{p}. \tag{2}$$

We use only the visible markers and bundles at each frame, so the number of constraints may vary across frames. A typical frame contains up to 50 3*D* constraints.

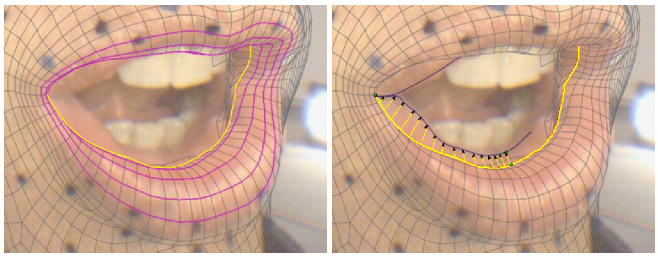
4.2 Contour constraints

Our input tracked curves are primary contours on the face: inner lip and eyelids that follow the silhouette edges. Unlike fixed contours such as outer lips or nasolabial folds, the primary contours are free to slide on the mesh surface. For example, the silhouette of the inner mouth changes when the lip rolls out. We want to dynamically associate each tracked curve to an *edge-contour* on the polygonal mesh at each frame. An edge-contour contains a subset of connected edges from an *edge-loop* on the mesh. Figure 2(a) shows several candidate edge-loops in the mouth region.

Contour to curve correspondence. To match a tracked curve on the face, we choose the edge contour on the geometry that has the maximum number of silhouette edges as seen from the cameras (yellow contour in Figure 2(a)). A silhouette edge is defined as an edge shared by a visible polygon and a hidden polygon. We refer to this selected edge contour as an *occluding contour*.

The correspondence between the occluding contour and the tracked curve is initialized from the mesh of the previous frame. Because the optimal solution may change the occluding contour, we can iterate between finding correspondence and solving optimization until convergence. In practice, we find one iteration sufficient for all our examples. By allowing the occluding contours to change across frames, we can continuously adapt the lip geometry to the tracked curves during complex lip motion.

Fitting 2*D* curves. Once an occluding contour is selected to match a tracked curve, we create a mapping between the two. We first project all of contour vertices to the image plane, then we align the end-points of the occluding contours to the corresponding end-points on the tracked curves. In our current setup we have four tracked curves in the mouth area: separate curves for lower and upper lips from each camera. Therefore we split the occluding contour into four segments using the two lip corners and the middle of the lips as the end points. For each end point on the occluding contour, we pick the closest projection of the tracked curve as the corresponding end point. Finally, we use arc-length based mapping to establish correspondences between the interior points (Figure 2(b)). As a result, Equation (1) for fitting 2*D* markers can be directly applied to fit the occluding contour's vertices to the tracked curve.



(a) Edge loops on the lip. (b) Contour-curve correspondence.

Figure 2: *Correspondence between an occluding contour and a tracked curve.*

Fitting 3*D* eyelid curves. Using visibility information helps reducing the ambiguity stemming from using 2*D* constraints. Similarly, we can also incorporate other prior knowledge of the tracked curves into our framework. Specifically, we utilize the fact that eyelids slide on the surface of the cornea, which is a relatively rigid object. We incorporate this prior by projecting the eyelid curves onto the cornea from the camera to obtain a 3*D* curve. Using the corresponding points computed on the tracked curve, we can apply Equation (2) to fit the resulting 3*D* curve.

4.3 Optimization framework

We collect at every frame all the 2*D* marker constraints and 3*D* bundle constraints, including those from the tracked curves, to formulate the optimization problem. The energy we minimize is the squared *l*2 error of all the constraints.

Blendshape fitting. The first optimization solves for blendshape weights $\mathbf{w}$ that best match the tracked markers and curves. Given a neutral mesh $\mathbf{V}_0$ and the blendshape basis $\mathbf{B}$ with each blendshape as a column, the goal is to fit the deformed mesh $\mathbf{V}(\mathbf{w}) = \mathbf{V}_0 + \mathbf{B}\mathbf{w}$ to the input features.

$$\min_{\mathbf{w}} \sum_{i=1}^{2} \omega_i E_i(\mathbf{w}) \qquad \text{s.t.} \qquad -\mathbf{1} \le \mathbf{w} \le \mathbf{1}\,. \tag{3}$$

The two energies terms are 2*D* markers and 2*D* curves of the eyes and the inner mouth respectively, with weights $\omega = 1.0$. Since both constraints are linear to the blendshape weights, we solve this problem using quadratic programming to enforce the bounds. The result is a deformed mesh, $\mathbf{V}$, in the blendshape subspace.

Out-of-subspace corrective. Depending on the quality of the blendshapes, $\mathbf{V}$ may not capture the given performance accurately. We want to better match the constraints by solving for an additional

corrective shape, $\Delta\mathbf{V}$, without inducing noise or over-fitting. This optimization has five energy terms. In addition to the $2D$ markers and $2D$ curves, we also use $3D$ curves for the eyelids. To prevent arbitrary changes to the mesh geometry, we use a cotangent weighted Laplacian constraint [Meyer et al. 2002] for regularization. The mesh boundary between the deformable portion of the face and the rest of the body mesh is used as $3D$ point constraints, and it remains fixed for the entire animation.

$$\min_{\Delta\mathbf{V}} \sum_{i=1}^{5} \omega_i E_i(\Delta\mathbf{V}). \tag{4}$$

In our experiments, we use $\omega = 1.0$ for the Laplacian constraint, the bundles, and the markers; and $\omega = 0.5$ for constraints associated with contours. Because all constraints are linear, we can solve the corresponding linear system with QR factorization to obtain the least-square solution.

The final shape $\tilde{\mathbf{V}}$ combines solutions from both optimizations: $\tilde{\mathbf{V}} = \mathbf{V} + \Delta\mathbf{V}$.

5 Results

Figure 3: Digital double facial animation of two actors captured by our system.

Our system has been applied in game cinematics to create several minutes of hero facial animation captured from two actors (Figure 3). We used a facial rig with 60 blendshapes as input to the solver framework. The facial mesh contains 7800 vertices and the overall processing time is a few seconds per frame, including retargeting, on a mainstream PC. Our experiments demonstrate a visible improvement in the quality of performance reconstruction and retargeting due to contour integration when compared against state-of-the art facial capture solutions.

We compare the quality of the fit at different stages of the reconstruction pipeline in Figure 4 . Solving for blendshapes (Figure 4b) results in a good overall match and gets the outer lips to match correctly. Adding contours into the blendshape solve improves the results around the eyelids (Figure 4c) but is not accurate enough to capture the shape of the inner mouth. Solving for out-of-subspace correctives (Figure 4d) successfully captures the inner lip shape, including the corner of the mouth. Another technique for measuring the tracking accuracy [Bradley et al. 2010] is presented in Figure 5. The two frames shown here are taken from a 2d texture image sequence obtained by projecting the helmet camera images onto the texture space of the actor's mesh. The reprojected texture sequence remains relatively fixed through the sequence even though the 3d mesh undergoes significant changes in deformation, including relatively large motions of the mouth.

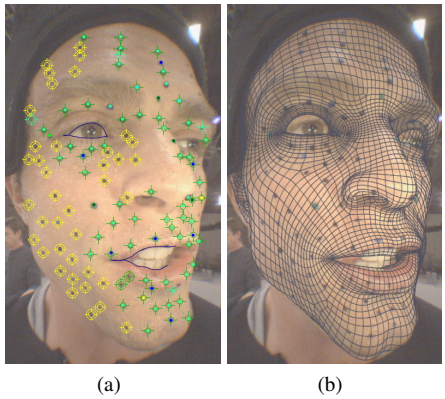


Figure 4: Contour integration improves expression of eyelids and mouth. *(a) Solver input: green and blue dots are 3D bundles, yellow dots are 2D markers and the blue curves are the contours tracked from this camera. (b) Blendshape solve without using contours. (c) Blendshape solve with contours, note the improvement in the eyelid and upper lip areas. (d) Corrective shape captures inner-lip and mouth corner regions accurately.*

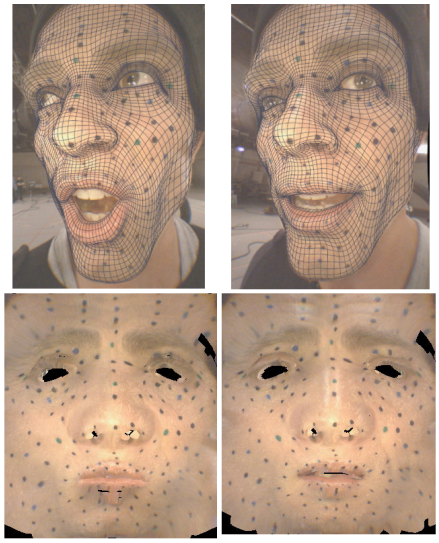


Figure 5: Validating tracking accuracy with 2d texture reprojection. *Top row shows two frames with different facial expressions (200 frames apart), and the bottom row shows corresponding 2d texture images obtained by projecting the input images on the texture coordinates of the tracked mesh.*

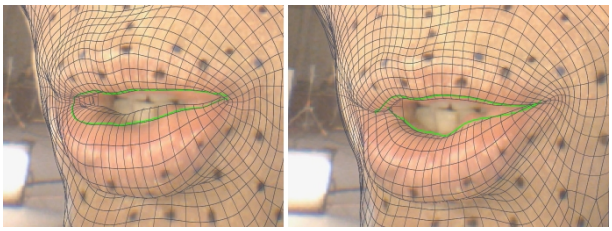


Figure 6: Using sliding occlusion information improves lip shape. *(a) Fixed correspondences causes the lower lip to bulge incorrectly. (b) Sliding occluding contours update yields correct correspondence and captures lip shape correctly.*

We highlight the importance of incorporating occluding contours into the reconstruction framework in Figure 6. Matching a fixed edge contour to the tracked 2D curve causes the optimizer to produce incorrect deformations when the actor's lip undergoes complex roll motions (Figure 6a). Using a sliding occluding contour, which is updated for each frame, (Figure 6b) allows the optimizer to match the tracked curve successfully.

Our results show that using primary occluding contours and dot features produces a very accurate match of the overall face. The tracked facial geometry also matches the secondary contours such as nasolabial folds even though these contours are not explicitly included in the optimization framework.

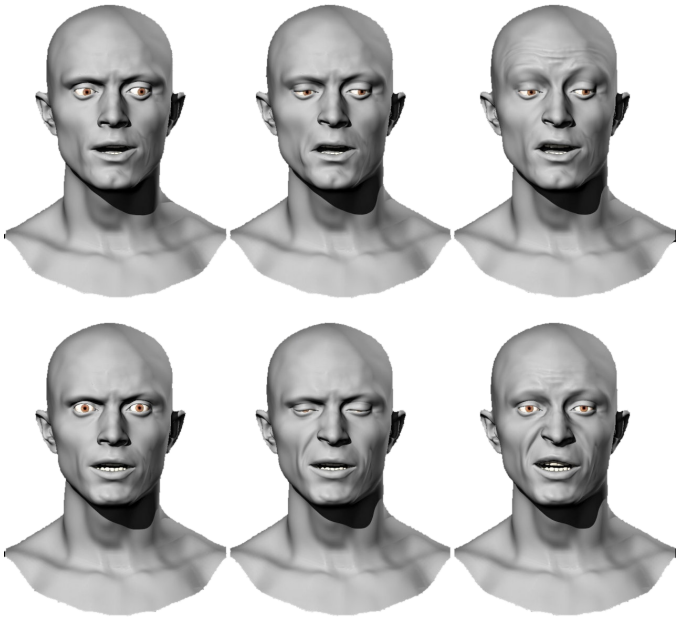


Figure 7: Performance capture results are editable. *Left: original expression; center: modified blendshape weights; right: resculpted shapes.*

A distinguishing feature of our facial capture pipeline is editability, which allowed us to deploy this system in production. Figure 7 shows the solved performance on the left column along with results of two common editing operations shown in the next two columns. Because the majority of the facial deformation is captured in the blendshapes, an animator can adjust the blendshape curves to create an edited performance without losing the details from the solve. An example of curve editing is shown in the middle column, where we have added extra eye blinks and raised upper the lip line. In other production scenarios, it may be useful to resculpt the input blendshapes after the solve. For example, the right column shows the extra wrinkle details added to the eyebrow-raise blendshape along with edits to sneer and nasolabial furrow. So, whenever the actor raises his eyebrow in this performance, we will notice the extra wrinkles appearing in the edited performance. This gives artists the freedom to update the actor blendshape model while retaining the performance from the original solve.

5.1 Animation retargeting

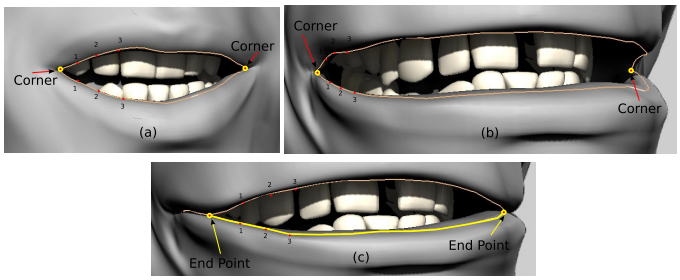


Figure 8: Heuristic for retargeting lip shapes. *(a) Actor's lip shape with corners and corresponding vertices on upper and lower occluding contours. (b) Retargeting onto creature with large mouth sculpt does not capture the inner lip shape because the existing correspondences are not semantically meaningful. (c) Choosing a smaller* talking *portion of the lip contour for retargeting by choosing new end points produces a nicer match to the input shape. New correspondences are computed based on the end points.*

We developed a retargeting system to animate the facial performance of a non-human creature using the solved data on the actor's rig. First, we create a blendshape rig for the creature by transferring the actor's blendshapes using a deformation transfer algorithm [Sumner and Popović 2004]. This creates parallel blendshape rigs for the actor and creature [Sagar 2006]. We synthesize the creature animation by copying the blendshape weights from the actor's solve. This re-creates the performance on the creature geometry, which can be creatively adjusted by an artist, if desired, by directly editing the blendshape curves or using spacetime editing techniques [Seol et al. 2012]. Next, we retarget the correctives from the actor which improves the overall quality of retargeted performance and captures subtle skin movements. The combination of blendshape retargeting with correctives produces compelling lifelike movement for most regions of the face. We show that the actors' facial expressions are captured, reproduced and retargeted to the creature at high-fidelity in Figure 9.

However, when the creature mouth geometry is significantly larger than the actor's mouth, this strategy fails to produce a lip readable animation on the creature. This is primarily because the semantic correspondence between the points on the inner lips is lost when the mouth shape is resculpted to the creature's mouth. This issue may rise when retargeting to wide mouthed creatures, for example sharks or other reptile creatures.

After experimenting with several strategies, we discovered that the actor's lip shapes work well when the length of the *talking* portion of the creature's inner lip contours approximately matches the length of the actor's inner lips (Figure 8). We adjust the end points on the creature's inner lip contours and establish correspondences

Figure 9: *Performance capture and retargeting. The actor expressions are carried over to his avatar and to retargeted to the creature.*

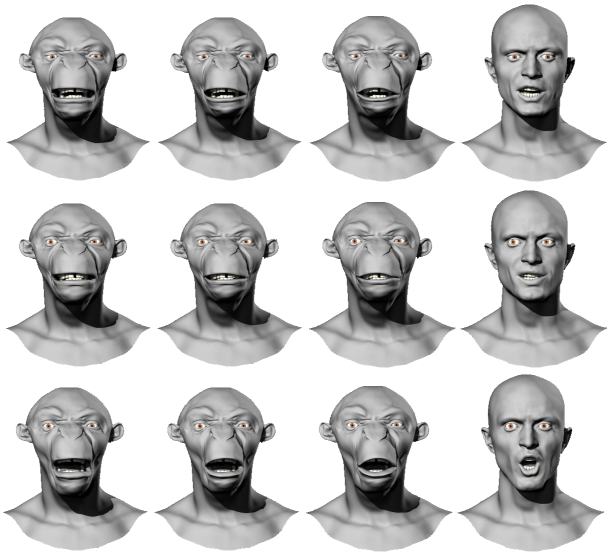


Figure 10: Lips retargeting. *The left column shows retargeting without the lip sync deformer and the next two columns have two different threshold settings that produce different lip shapes during retargeting. The right column shows the corresponding source shapes on the actors mesh.*

with the corners of the actor's lips. Once the end points are associated, we use an algorithm similar to 4.2 to compute the correspondences between the interior vertices along the occluding contours of the actor and creature. Subsequently, we apply the out-of-subspace deformation algorithm (section 4.3) to compute the lip-sync shape.

The above computation uses an additional constraint that enforces the distance between points on the upper and lower lips on the creature's lip contours to be the same distance as the actor's, for the corresponding points. We also constrain all the points on the creature contour between the original lip corner and the new end points to have zero length.

In practice, we use a distance thresholding scheme to automatically select the end points on the creature's lip contours. Our system also lets animators adjust the end points on specific frames to fine-tune the inner lip shape. Figure 10 shows several frames of retargeting the actor's performance with different settings of the end points. These results and the accompanying video shows that our system is capable of generating a range of retargeted lip shapes while matching the overall performance of the actor.

6 Discussion

We have presented a novel method that captures important nuanced performances that are very difficult to capture using previous methods. We achieve this by integrating contours and applying a two layered animation approach (blendshape and per-frame correctives).

Contours are currently tracked manually, although our system can work with automatically tracked contours. Automatic tracking is a daunting task, given the challenge of accurately tracking the inner lip contours – they are too easily confused against the teeth, gums, and tongue. Eye contours are slightly easier, but suffer from specular highlights, texture variations, and occlusions caused by eyelashes. Manual tracking is a relatively simple task for rotoscope artists (tracing curves on an image), but it comes at a significant added production cost. As future work, we plan to use the manually tracked contours to bootstrap vision algorithms for robust automatic contour-tracking.

The examples presented in this paper use footage from head-mounted cameras, which provides optimal data for simultaneous capture of facial and body motions. Our techniques generalizes to other performance capture scenarios, such as a static bank of cameras or handheld witness cameras following the actor's face.

Our system is not limited to blend-shape rigs, and we plan to extend further to include joints and other deformation techniques. For future work, we would like to directly incorporate the correctives into the blendshape rig during the solve, thus improving the quality of the tracking rig.

Without contours and corrective shapes integrated into the captured performance, the quality of the result is not good enough to be used for high-end production. To the best of our knowledge, prior facial capture methods require that the blendshapes be hand-processed by skilled animators. Additionally, to achieve final quality results, a modeler sculpts per-frame corrective shapes on top of the edited animation. These edits compromise accuracy in exchange for aesthetics or other subjective criteria. Our technique focuses on perceptually relevant features and matches them very accurately, delivering a captured performance that needs minimal aesthetic edits.

To be able to meaningfully edit the results, a high quality blendshape library is needed. Although our system can work without a blendshape rig, the resulting correctives are hard to modify and edit. A promising direction for future research would be to adapt

a generic facial blendshape rig to the actor's anatomical structure as a basis for tracking. The blendshape library is also useful in our retargeting techniques, allowing for the resulting retargeted performance to be modified by an artist. Pursuing other strategies for rig retargeting is another promising avenue of research, allowing for higher quality performance transfer.

References

ALEXANDER, O., ROGERS, M., LAMBETH, W., CHIANG, J.-Y., MA, W.-C., WANG, C.-C., AND DEBEVEC, P. 2010. The digital emily project: Achieving a photorealistic digital actor. *Computer Graphics and Applications, IEEE 30*, 4 (july-aug.), 20 –31.

BALTRUSAITIS, T., ROBINSON, P., AND MORENCY, L.-P. 2012. 3D constrained local model for rigid and non-rigid facial tracking. In *Computer Vision and Pattern Recognition (CVPR 2012)*.

BASCLE, B., AND BLAKE, A. 1998. Separability of pose and expression in facial tracking and animation. In *Computer Vision, 1998. Sixth International Conference on*, IEEE, 323–328.

BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W., AND GROSS, M. 2011. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph. 30* (August), 75:1–75:10.

BICKEL, B., BOTSCH, M., ANGST, R., MATUSIK, W., OTADUY, M., PFISTER, H., AND GROSS, M. 2007. Multi-scale capture of facial geometry and motion. *ACM Trans. Graph. 26*, 3 (July).

BLACK, M. J., AND YACOOB, Y. 1995. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proceedings of the Fifth International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, USA, ICCV '95, 374–.

BORSHUKOV, G., PIPONI, D., LARSEN, O., LEWIS, J. P., AND TEMPELAAR-LIETZ, C. 2005. Universal capture - image-based facial animation for "the matrix reloaded". In *ACM SIGGRAPH 2005 Courses*, ACM, New York, NY, USA, SIGGRAPH '05.

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (Jan.), 213–230.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. *ACM Trans. Graph. 29*, 4 (July), 41:1–41:10.

BREGLER, C., BHAT, K., SALTZMAN, J., AND ALLEN, B. 2009. Ilm's multitrack: a new visual tracking framework for high-end vfx production. In *SIGGRAPH 2009: Talks*, ACM, New York, NY, USA, SIGGRAPH '09, 29:1–29:1.

CHUANG, E., AND BREGLER, C. 2002. Performance driven facial animation using blendshape interpolation. Tech. rep., Stanford University.

COOTES, T. F., EDWARDS, G. J., AND TAYLOR, C. J. 1998. Active appearance models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Springer, 484–498.

COVELL, M., AND BREGLER, C. 1996. Eigen-points. In *Image Processing, 1996. Proceedings., International Conference on*, vol. 3, IEEE, 471–474.

CUNNINGHAM, D. W., KLEINER, M., WALLRAVEN, C., AND BÜLTHOFF, H. H. 2005. Manipulating video sequences to determine the components of conversational facial expressions. *ACM Trans. Appl. Percept. 2*, 3 (July), 251–269.

DIMENSIONAL IMAGING LTD. DI4D^{TM}. `http://www.di3d.com/`.

DYNAMIXYZ. HMC & Grabber^{TM}. `http://www.dynamixyz.com`.

EKMAN, P. 1993. Facial expression and emotion. *American Psychologist 48*, 4, 384–392.

FYFFE, G., HAWKINS, T., WATTS, C., MA, W.-C., AND DEBEVEC, P. 2011. Comprehensive facial performance capture. In *Eurographics 2011*.

GHOSH, A., FYFFE, G., TUNWATTANAPONG, B., BUSCH, J., YU, X., AND DEBEVEC, P. 2011. Multiview face capture using polarized spherical gradient illumination. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*.

HORN, B., AND SCHUNCK, B. 1981. Determining optical flow. *Artificial intelligence 17*, 1, 185–203.

HUANG, H., CHAI, J., TONG, X., AND WU, H.-T. 2011. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. *ACM Trans. Graph. 30*, 4 (July), 74:1–74:10.

IMAGE METRICS. Live Driver^{TM}. `http://www.image-metrics.com`.

LAU, M., CHAI, J., XU, Y.-Q., AND SHUM, H.-Y. 2009. Face poser: Interactive modeling of 3d facial expressions using facial priors. *ACM Trans. Graph. 29*, 1 (Dec.), 3:1–3:17.

LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2009) 28*, 5.

LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2010) 29*, 3 (July).

LI, H., YU, J., YE, Y., AND BREGLER, C. 2013. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics (Proc. SIGGRAPH) 32*, 4.

LIU, X., MAO, T., XIA, S., YU, Y., AND WANG, Z. 2008. Facial animation by optimized blendshapes from motion capture data. *Computer Animation and Virtual Worlds 19*, 3–4, 235–245.

MA, W.-C., JONES, A., CHIANG, J.-Y., HAWKINS, T., FREDERIKSEN, S., PEERS, P., VUKOVIC, M., OUHYOUNG, M., AND DEBEVEC, P. 2008. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*.

MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2002. Discrete differential-geometry operators for triangulated 2-manifolds. In *Proc. VisMath*, 35–57.

MIRANDA, J. C., ALVAREZ, X., ORVALHO, J. A., GUTIERREZ, D., SOUSA, A. A., AND ORVALHO, V. 2011. Sketch express: facial expressions made easy. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, ACM, New York, NY, USA, SBIM '11, 87–94.

MOVA. CONTOUR^{TM}Reality Capture. `http://www.mova.com/`.

NUSSECK, M., CUNNINGHAM, D. W., WALLRAVEN, C., AND BÜLTHOFF, H. H. 2008. The contribution of different facial

regions to the recognition of conversational expressions. *Journal of Vision 8*, 8.

ORVALHO, V., BASTOS, P., PARKE, F., OLIVEIRA, B., AND ALVAREZ, X. 2012. A facial rigging survey. In *in Proc. of the 33rd Annual Conference of the European Association for Computer Graphics - Eurographics*, ACM, 10–32.

PIGHIN, F. H., SZELISKI, R., AND SALESIN, D. 1999. Resynthesizing Facial Animation through 3D Model-based Tracking. In *Proc. 7th International Conference on Computer Vision, Kerkyra, Greece*, 143–150.

SAGAR, M. 2006. Facial performance capture and expressive translation for king kong. In *ACM SIGGRAPH 2006 Sketches*, ACM, New York, NY, USA, SIGGRAPH '06.

SEO, J., IRVING, G., LEWIS, J. P., AND NOH, J. 2011. Compression and direct manipulation of complex blendshape models. *ACM Trans. Graph. 30*, 6 (Dec.), 164:1–164:10.

SEOL, Y., LEWIS, J., SEO, J., CHOI, B., ANJYO, K., AND NOH, J. 2012. Spacetime expression cloning for blendshapes. *ACM Trans. Graph. 31*, 2 (Apr.), 14:1–14:12.

SIFAKIS, E., NEVEROV, I., AND FEDKIW, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. 24*, 3 (July), 417–425.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (Aug.), 399–405.

TERZOPOULOS, D., AND WATERS, K. 1993. Analysis and synthesis of facial image sequences using physical and anatomical models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 15*, 6, 569–579.

WEISE, T., LI, H., GOOL, L. V., AND PAULY, M. 2009. Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)*, Eurographics Association, ETH Zurich.

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2011) 30*, 4 (July).

WILLIAMS, L. 1990. Performance-driven facial animation. *SIGGRAPH Comput. Graph. 24*, 4 (Sept.), 235–242.

ZHANG, S., AND HUANG, P. 2004. High-resolution, real-time 3d shape acquisition. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 3 - Volume 03*, IEEE Computer Society, Washington, DC, USA, CVPRW '04, 28–.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conference on Computer Graphics*, 548–558.